On Adaptive Multiple-Shooting Method for Stochastic Multi-Point Boundary Value Problems

Ali Foroush Bastani*   Davood Damircheli

Department of Mathematics, Inistitute for Advanced Studies in Basic Sciences, P.O. Box 45195-1159, Zanjan, Iran

Abstract This paper presents an adaptive multiple-shooting method to solve stochastic multi-point boundary value problems. The heuristic to choose the shooting points is based on separating the effects of drift and diffusion terms and comparing the corresponding solution components with a pre-specified initial approximation. Having obtained the mesh points, we solve the underlying stochastic differential equation on each shooting interval with a first-order strongly-convergent stochastic Runge-Kutta method. We illustrate the effectiveness of this approach on 1-dimentional and 2-dimentional test problems and compare our results with other non-adaptive alternative techniques proposed in the literature.

1 Introduction

Numerical methods for solving initial value problems in stochastic differential equations (SDE-IVPs) have been extensively researched in the last two decades (see e.g. [17, 20] and the references therein). This is not the case for stochastic boundary value problems (SDE-BVPs or SBVPs for short), because of complications both in theoretical as well as computational aspects. These equations appear naturally in a variety of fields such as smoothing [26], maximum a posteriori estimation of trajectories of diffusions [30], wave motion in random media [14], stochastic optimal control [31], valuation of boundary-linked assets [11] and in the study of reciprocal processes [18]. They also arise from the semi-discretization in space of stochastic partial differential equations by the method of lines approximation [19]. Taking into account the fact that the exact solution of these equations are rarely available in analytic form, trying to find efficient approximation schemes for the trajectories of the solution process or its moments, seems to be a natural candidate. During the last years, several authors have studied with different techniques, the numerical

---

*Corresponding author

solution of SBVPs of the form:

$$\begin{cases} dX(t) = f(X(t), t)dt + g(X(t), t) \circ dW(t), & X(t) \in \mathbb{R}^d, \quad 0 \le t \le T, \\ \alpha(X) = c, \end{cases} \tag{1.1}$$

in which $f : \mathbb{R}^d \times [0, T] \rightarrow \mathbb{R}^d$ and $g : \mathbb{R}^d \times [0, T] \rightarrow \mathbb{R}^{d \times d}$ are continuous globally Lipschitz functions with polynomial growth, $W(t)$ is a d-dimensional Wiener process, $\alpha : C^0(\mathbb{R}^d \times [0, T]) \rightarrow \mathbb{R}^d$ is a continuous operator and $c \in \mathbb{R}^d$ is a constant vector. The existence and uniqueness of the solution process as well as the **Markov field property** of it have been studied by some authors, among them we mention [24, 25, 23, 31, 15]. Due to the anticipative nature of the solution process, the main machinery in the study of these equations have turned out to be the Malliavin calculus [21].

The majority of research in this field has concentrated around two-point SBVPs (TP-SBVPs) corresponding to the choice

$$\alpha(X) = h(X(0), X(T)) = c, \tag{1.2}$$

in which $h : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a given (possibly nonlinear) function and c is defined as before. In this category, we must point out to **linear TP-SBVPs** in which both the drift and diffusion coefficients (f and g respectively in (1.1)) are linear functions of their arguments and the function h is of the form

$$h(y, z) = H_0 y + H_1 z, \tag{1.3}$$

in which H_0 and H_1 are $d \times d$ matrices. At the same time, the special class of **functional boundary conditions** of the form

$$\alpha(X) = \int_0^T dA(t)X(t) = c, \tag{1.4}$$

have also been of interest, in which $A(t)$ is a $d \times d$ matrix valued integrator. The other interesting case is the multi-point SBVP (or MP-SBVP for short) having the boundary condition

$$\alpha(X) = \sum_{j=1}^{N_s} A_j X(\tau_j) = c, \tag{1.5}$$

in which $A_1, A_2, \cdots, A_m$ are constant square matrices of order d and $\tau_1, \tau_2, \cdots, \tau_{N_s} \in [0, T]$ are given *switching points* with the property $\tau_i < \tau_j$, for $i < j$. This boundary condition could be considered as the result of a quadrature formula applied to approximate the general form (1.4) and will be of special interest in this paper.

On the numerical side, some efforts have been directed towards devising efficient numerical schemes for (1.1) among them we mention the following: Allen and Nunn [3] propose two methods for linear two-dimensional second order SBVPs, one based on finite differences and the other based on simple-shooting. They analyze the convergence properties of these methods and report some numerical experiments confirming their theoretical results. Arciniega and Allen [5] examine a shooting-type method for systems of linear SBVPs of the form (1.1). This method could be viewed as a generalization of the *complementary function approach* for deterministic BVPs adopted to solve SBVPs [28]. Arciniega [4] extends this work to the nonlinear case and performs some error analysis for this new scheme. Ferrante, Kohatsu-Higa and Sanz-Solé [12] use a strong Euler-Maruyama approximation to find strong solutions of (1.1) with linear boundary conditions. They obtain error

estimates for this method without accompanying any numerical results to their theoretical findings. In a recent paper, Esteban-Bravo and Vidal-Sanz [11] use the wavelet-collocation scheme to find approximations to trajectories of the solution for a general version of (1.1) with boundary conditions of the form (1.4). We must also mention the work of Prigarin and Winkler [27] in which they propose a special member of the general Markov chain Monte Carlo (MCMC) approach namely the Gibbs sampler to construct realizations of the solution process. The convergence is proved for the special case of linear TP-SBVPs and some guidelines have been provided to cope with the general nonlinear case and also boundary value problems for stochastic partial differential equations.

Among the above-mentioned schemes, the simple-shooting method which relies on transforming the SBVP (1.1) to an SDE-IVP, has shown to have good accuracy properties, but it may give unacceptable approximate solutions on long time intervals. This is specially the case when the underlying SDE is *unstable* i.e. almost all sample paths are rapidly growing in absolute value. Our aim here is to circumvent this deficiency by developing an adaptive multiple-shooting method to solve (1.1) based on a detailed analysis of the sample paths of the corresponding stochastic equation. The idea is to adaptively subdivide the typical interval $[\tau_i, \tau_{i+1}]$ into a grid of *shooting points*

$$\tau_i = t_{i,1}(\omega) < t_{i,2}(\omega) < \cdots < t_{i,j}(\omega) < \cdots < t_{i,N(i)}(\omega) = \tau_{i+1}$$

in which the $t_{i,j}$'s and also $N(i)$ will depend on the particular realization (indexed by ω) of the underlying Wiener process. In each interval $[\tau_i, \tau_{i+1}]$, starting from $t_{i,1} = \tau_i$, the criterion we choose to obtain $t_{i,j+1}$ from $t_{i,j}$ is to use an idea adopted from the **operator-splitting** method to investigate the behavior of the two local SDE-IVPs arising from the drift and diffusion components of the underlying SDE and controlling upon their growth on this subinterval. For this purpose, we employ an initial approximation to the solution which (approximately) satisfies the boundary conditions and compare it with the two corresponding SDE-IVP solutions. To obtain the mesh points, we solve the above mentioned SDE-IVPs on each shooting interval with a first-order strongly convergent **stochastic Runge-Kutta** method introduced in [13]. We show that this strategy significantly enhances the accuracy and stability properties of the simple-shooting method and at the same time reduces the computational cost of the long-time integration problem to a great extent. Comparison with other schemes like simple-shooting, finite-differences, wavelet-collocation and also the fixed-step multiple shooting method itself, confirms that the proposed method is a reliable alternative than the widely used non-adaptive approaches in the literature.

The rest of this paper is organized as follows. In section 2, we present the multiple shooting framework to solve SBVPs with multi-point boundary conditions. The criterion to select the shooting points which forms the foundation of our adaptive strategy will be discussed in section 3. The details of optimal parameter tuning for the proposed scheme and implementation details will be described in section 4. We conclude the paper by commenting on some possible ways to extend this work into more general frameworks.

2 Multiple Shooting Method for MP-SBVPs

In this section, we describe the multiple-shooting framework to approximate the sample paths of the equation (1.1). This can be considered as the extension of methods presented

in [5, 4] and will serve as the ground base for our adaptive scheme. For this purpose, consider the following MP-SBVP in Stratonovich form:

$$\begin{cases} dX(t) = f(X(t),t)dt + g(X(t),t) \circ dW(t), & X(t) \in \mathbb{R}^d, \quad 0 \le t \le T, \\ \sum_{j=1}^{N_s} A_j X(\tau_j) = c. \end{cases} \tag{2.1}$$

Without loss of generality, we assume throughout the paper that $\tau_1 = 0$ and $\tau_{N_s} = T$. For each realization of the Wiener process, we are interested in finding the corresponding realization of the solution process satisfying (2.1). Assume that for each $i = 1, 2, \cdots, N_s - 1$, $I_i = [\tau_i, \tau_{i+1}]$ is subdivided into the shooting intervals $[t_{i,j}, t_{i,j+1}]$, $j = 1, \cdots, N(i) - 1$ with $t_{i,1} = \tau_i$ and $t_{i,N(i)} = \tau_{i+1}$. The adaptive procedure used to obtain them will be discussed in section 3 but in the sequel, we assume that they are known. If $X_{i,j}(t; s_{i,j})$ solves the local SDE-IVPs:

$$\begin{cases} dX_{i,j}(t; s_{i,j}) = f(X_{i,j}(t; s_{i,j}), t)dt + g(X_{i,j}(t; s_{i,j}), t) \circ dW(t), & t \in [t_{i,j}, t_{i,j+1}], \\ X_{i,j}(t_{i,j}; s_{i,j}) = s_{i,j}, \end{cases} \tag{2.2}$$

for $i = 1, 2, \cdots, N_s - 1$ and $j = 1, \cdots, N(i) - 1$, augmentation of the local solutions $X_{i,j}(t; s_{i,j})$ with imposition of continuity condition at the interior shooting points and satisfaction of multi-point boundary conditions at switching points will result in a global approximation to $X(t)$. For this purpose, we find the unknown initial conditions $s_{i,j}$'s by solving the system of $D = d \times (\sum_{i=1}^{N_s-1}[N(i) - 1]) + 1$ nonlinear equations:

$$F(\mathbf{s}) = 0, \tag{2.3}$$

in which

$$\mathbf{s} = (s_{1,1}^T, \cdots, s_{1,N(1)-1}^T, s_{2,1}^T, \cdots, s_{2,N(2)-1}^T, \cdots, s_{N_s-1,1}^T, \cdots, s_{N_s-1,N(N_s-1)-1}^T, s_{N_s,1}^T)^T \in \mathbb{R}^D,$$

is the *shooting vector* and $F(\mathbf{s})$ is given by:

$$F(\mathbf{s}) = \begin{bmatrix} \mathbf{s}_2 - \mathbf{X}_1(\mathbf{s}_1) \\ \vdots \\ \mathbf{s}_{N_s} - \mathbf{X}_{N_s-1}(\mathbf{s}_{N_s-1}) \\ g(\mathbf{s}, c) \end{bmatrix}, \tag{2.4}$$

in which $\mathbf{s}_j = (s_{j,2}, \ s_{j,3}, \cdots, s_{j,N(j)-1}, \ s_{j+1,1})^T$ for $j = 2, 3, \cdots, N_s$,

$$\mathbf{X}_j = \begin{bmatrix} X_{j,1}(t_{j,2}; s_{j,1}) \\ X_{j,2}(t_{j,3}; s_{j,2}) \\ \vdots \\ X_{j,N(j)-2}(t_{j,N(j)-1}; s_{j,N(j)-2}) \\ X_{j,N(j)-1}(t_{j+1,1}; s_{j,N(j)-1}) \end{bmatrix}, \tag{2.5}$$

for $j = 1, \cdots, N_s - 1$ and

$$g(\mathbf{s}, c) = A_1 s_{1,1} + A_2 s_{2,1} + \cdots + A_{N_s-1} s_{N_s-1,1} + A_{N_s} s_{N_s,1} - c. \tag{2.6}$$

The solution of system (2.4), which provides a global refinement of the solution values at the gridpoints, is usually done within the framework of a damped-Newton iteration whose k-th iteration $\mathbf{s}^k$ is of the form

$$\mathbf{s}^{k+1} = \mathbf{s}^k - \lambda_k [DF(\mathbf{s}^k)]^{-1} F(\mathbf{s}^k). \tag{2.7}$$

In this relation, $\lambda_k \in (0,1]$ is the *relaxation* or *damping factor* and $DF(\mathbf{s}^k)$ is the Jacobian matrix of $F(\mathbf{s})$ evaluated at the k-th iteration.

It can be shown that

$$DF(\mathbf{s}) = \begin{bmatrix} -\mathbf{\Gamma}_1 & \mathbf{I}_1 & & & \\ & -\mathbf{\Gamma}_2 & \mathbf{I}_2 & & \\ & & \ddots & \ddots & \\ & & & -\mathbf{\Gamma}_{N_s-1} & \mathbf{I}_{N_s-1} \\ A_1 & A_2 & \dots & A_{N_s-1} & A_{N_s} \end{bmatrix}$$

in which

$$\mathbf{\Gamma}_j = \begin{bmatrix} \Gamma_{j,1} & & & \\ & \Gamma_{j,2} & & \\ & & \ddots & \\ & & & \Gamma_{j,N(j)} \end{bmatrix} \tag{2.8}$$

and the components $\Gamma_{j,k} \equiv D_{s_{j,k}} X_{j,k-1}(t_{j,k}; s_{j,k-1})$ for each j and k are $d \times d$ matrices and

$$\mathbf{I}_j = \begin{bmatrix} I_{d \times d} & & & \\ & I_{d \times d} & & \\ & & \ddots & \\ & & & I_{d \times d} \end{bmatrix} \tag{2.9}$$

is an $N(j) \times N(j)$ identity matrix. It is obvious that the exact computation of $\Gamma_{j,k}$ requires the analytic solution of the local SDE-IVPs (2.2). It is worth pointing out here that although it is possible to approximate $\Gamma_{j,k}$'s by linearization of the corresponding local SDEs and integrating them up to $t_{i,j}$, we will adopt an alternative strategy by approximating the derivative terms by finite differences. A strategy for choosing the λ_k's has also been developed and thoroughly tested in [9] which will be pursued here.

3   Adaptive Sequential Selection of Shooting Points

Multiple-shooting method as a natural generalization of the simple-shooting idea, significantly enhances the stability properties of its ancestor and behaves much better than it in terms of accuracy and rate of convergence and so has been a preferred choice to solve deterministic boundary value problems [6, 16, 29]. The main drawback of this method could be attributed to its computational cost which is directly proportional to the number of shooting points in the integration interval. To reduce these costs, some authors have proposed to devise a control mechanism on the number and location of shooting points in such a way that the stability and accuracy properties of the method are preserved. This strategy has the additional advantage of resolving the special features of the solution in the integration interval: "... a multiple-shooting approach should permit step sizes to be

chosen sequentially, fine in the boundary layers, and coarse in the smooth regions" [10]. We extend this argument to the case of non-smooth solutions - the feature which is intrinsic for SDEs - and show that the adaptive selection of shooting points based on the driving force for this non-smooth behavior, i.e. the underlying Wiener process and also comparing the solution with an initial approximate solution, will have an overall performance much better than the corresponding fixed step-size counterpart.

To start the adaptive procedure, we first find a simple piecewise linear approximation to the solution, $\theta(t)$, which approximately satisfies the multi-point boundary conditions (1.5). To find this approximation, we discretize the SDE-IVP in each interval with the Euler-Maruyama method and then solve the following system of nonlinear equations for $\overline{\theta}_{\tau_j}$'s:

$$
\begin{cases}
\overline{\theta}_{\tau_2} = \overline{\theta}_{\tau_1} + (\tau_2 - \tau_1)f(\overline{\theta}_{\tau_1}, \tau_1) + (W(\tau_2) - W(\tau_1))g(\overline{\theta}_{\tau_1}, \tau_1), \\
\overline{\theta}_{\tau_3} = \overline{\theta}_{\tau_2} + (\tau_3 - \tau_2)f(\overline{\theta}_{\tau_2}, \tau_2) + (W(\tau_3) - W(\tau_2))g(\overline{\theta}_{\tau_2}, \tau_2), \\
\vdots \qquad \vdots \\
\overline{\theta}_{\tau_{N_s}} = \overline{\theta}_{\tau_{N_s-1}} + (\tau_{N_s} - \tau_{N_s-1})f(\overline{\theta}_{\tau_{N_s-1}}, \tau_{N_s-1}) + (W(\tau_{N_s}) - W(\tau_{N_s-1}))g(\overline{\theta}_{\tau_{N_s-1}}, \tau_{N_s-1}), \\
\sum_{j=1}^{N_s} A_j \overline{\theta}_{\tau_j} = c.
\end{cases}
\tag{3.1}
$$

The continuous piecewise linear approximation could then be obtained by linear interpolation:

$$
\theta(t) = \frac{t - \tau_i}{\tau_{i+1} - \tau_i}\overline{\theta}_{\tau_{i+1}} + \frac{\tau_{i+1} - t}{\tau_{i+1} - \tau_i}\overline{\theta}_{\tau_i}, \quad t \in [\tau_i, \tau_{i+1}], \quad i = 1, \cdots, N_s.
\tag{3.2}
$$

Consider now the interval $[\tau_i, \tau_{i+1}]$ and put $t_{i,1} := \tau_i$. Starting from $t_{i,j}$ and to obtain the next shooting point in this interval, we integrate the following two local SDE-IVP's:

$$
\begin{cases}
d\widehat{X}(t) &= f(\widehat{X}(t), t)dt, \qquad t \in [t_{i,j}, \tau_{i+1}], \\
\widehat{X}(t_{i,j}) &= \theta(t_{i,j}),
\end{cases}
\tag{3.3}
$$

$$
\begin{cases}
d\widetilde{X}(t) &= g(\widetilde{X}(t), t) \circ dW(t), \qquad t \in [t_{i,j}, \tau_{i+1}], \\
\widetilde{X}(t_{i,j}) &= \theta(t_{i,j}),
\end{cases}
\tag{3.4}
$$

by deterministic and stochastic components of an SRK method, described in the next section. We will terminate the integration when we reach the first point in our discretization satisfying:

$$
t_{i,j} \leq s \leq \tau_{i+1}, \quad \|\widehat{X}(s)\| \geq L_1(s) \quad \text{or} \quad \|\widetilde{X}(s)\| \geq L_2(s),
\tag{3.5}
$$

in which $L_1(s)$ and $L_2(s)$ will be specified in the sequel. We then put $t_{i,j+1} := s$ and restart the integration of both (3.3) and (3.4) from $t_{i,j+1}$ using $\widehat{X}(t_{i,j+1}) = \widetilde{X}(t_{i,j+1}) = \theta(t_{i,j+1})$ as the initial guess. This procedure will be continued up until the point $t_{i,N(i)} = \tau_{i+1}$ is reached and then will be continued from the next shooting interval to finally arrive at T.

The first untold story in our description of the algorithm is the selection of the "stop-loss functions" $L_1(s)$ and $L_2(s)$ which control upon the location of our shooting points. The most intuitionistic proposal could be

$$
L_1(s) = \alpha\|\theta(s)\|, \quad L_2(s) = \beta\|\theta(s)\|
\tag{3.6}
$$

for some positive constants α and β (see e.g. [29] Section 7.3.6 for a similar idea in the case of deterministic BVPs). We could choose the α and β coefficients in (3.6) time-dependent

and find an empirical optimal relation for them, but our numerical experience shows that the gain in efficiency is not substantial. We have also tested other stopping criteria based only on the size of the increments of the Wiener process which has resulted in the selection of more shooting points but has not improved the accuracy in a comprehendible way. Another proposal is to find the first point s which simultaneously maximizes the following quantities:

$$\mathbb{P}\Big(\frac{\|\widehat{X}(s)\|}{\|\theta(s)\|} \geq \alpha\Big), \quad \mathbb{P}\Big(\frac{\|\widetilde{X}(s)\|}{\|\theta(s)\|} \geq \beta\Big)$$

for given positive α and β. The idea has led us to solve simple constrained stochastic programming problems in each step (with exact solutions for linear SBVPs) that needs further investigation and will be pursued in a forthcoming paper.

It is evident from the form of our adaptation criteria that in the case of weak driving noise process, we are controlling upon the size of the solution process and look at the first time at which the norm of the solution starts to deviate from the initial piecewise linear approximation. On the other hand, when the increments of the Brownian noise become large in some portions of the solution domain, we must finish the integration and select the current point as a suitable shooting point. In both of these scenarios, we must come back to the initial approximation $\theta(t)$ and continue the integration from the initial value $\theta(t_{i,j+1})$.

Having described the way in which we choose our shooting points for each realization, we now need to tell the other story about the time marching procedure to solve our SDE-IVPs resulting from the multiple shooting method, which will be discussed in the next section.

3.1 Stochastic Runge-Kutta Family

Among the many possible choices of methods to integrate the ODE-IVP and SDE-IVP problems in (3.3) and (3.4) (assuming w.l.o.g. that both equations are autonomous), we choose to work with a special member from the general class of stochastic Runge-Kutta (SRK) methods of the form

$$\begin{cases} \overline{\eta}_i & = \overline{X}_n + h\sum_{j=1}^{s} a_{ij} f(\overline{\eta}_j) + J_1 \sum_{j=1}^{s} b_{ij} g(\overline{\eta}_j), \quad i = 1, \ldots, s \\ \overline{X}_{n+1} & = \overline{X}_n + h\sum_{j=1}^{s} \alpha_j f(\overline{\eta}_j) + J_1 \sum_{j=1}^{s} \gamma_j g(\overline{\eta}_j), \end{cases} \tag{3.7}$$

in which $\overline{X}_n$ and $\overline{X}_{n+1}$ are approximations to $X(t_n)$ and $X(t_{n+1})$ respectively, $h = t_{n+1} - t_n$ and $J_1 = \int_{t_n}^{t_{n+1}} \circ dW(s) = W(t_{n+1}) - W(t_n)$. Here, $A = (a_{ij})_{i,j=1}^{n}$ and $B = (b_{ij})_{i,j=1}^{n}$ are $s \times s$ matrices with real elements and $\alpha^T = (\alpha_1, \ldots, \alpha_s)$ and $\gamma^T = (\gamma_1, \ldots, \gamma_s)$ are row vectors in $\mathbb{R}^s$. A typical member of this family could be represented by the Butcher tableau

A	B
α^T	γ^T

and according to the theory presented in [7], the highest possible order of strong (and also weak) convergence among all consistent choices for A, B, α and γ is one (see [17] for notions of strong and weak convergence in the SDE literature). In this work, we use a

three stage SRK method (dubbed R3) as the underlying numerical integrator which has the tableau

$$
\begin{array}{ccc|ccc}
0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 \\
0 & \frac{3}{4} & 0 & 0 & \frac{3}{4} & 0 \\
\hline
\frac{2}{9} & \frac{3}{9} & \frac{4}{9} & \frac{2}{9} & \frac{3}{9} & \frac{4}{9}
\end{array}
$$

and its deterministic and stochastic components are themselves valid numerical integration schemes [13]. More specifically, we integrate the IVP (3.3) with a method of the form

$$
\begin{cases}
\overline{\eta}_1 & = \overline{X}_n, \\
\overline{\eta}_2 & = \overline{X}_n + \frac{h}{2} f(\overline{\eta}_1), \\
\overline{\eta}_3 & = \overline{X}_n + \frac{3h}{4} f(\overline{\eta}_2), \\
\overline{X}_{n+1} & = \overline{X}_n + \frac{h}{9} (2f(\overline{\eta}_1) + 3f(\overline{\eta}_2) + 4f(\overline{\eta}_3)),
\end{cases}
\tag{3.8}
$$

and integrate the SDE (3.4) with another method having the form

$$
\begin{cases}
\overline{\eta}_1 & = \overline{X}_n, \\
\overline{\eta}_2 & = \overline{X}_n + \frac{J_1}{2} g(\overline{\eta}_1), \\
\overline{\eta}_3 & = \overline{X}_n + \frac{3J_1}{4} g(\overline{\eta}_2), \\
\overline{X}_{n+1} & = \overline{X}_n + \frac{J_1}{9} (2g(\overline{\eta}_1) + 3g(\overline{\eta}_2) + 4g(\overline{\eta}_3)).
\end{cases}
\tag{3.9}
$$

It is interesting to note here that the first scheme has third-order of convergence for a deterministic IVP and this will result in higher precision when we are faced with an SDE-BVP having a weak driving noise. On the other hand and for the second scheme, we have first-order of strong convergence for drift-free SDEs and when the drift is going to diminish in some portions of the problem domain, we have an exact-enough method to trace the non-smooth path of the corresponding realization.

4 Numerical Experiments

In this section, we report on the numerical results obtained using the adaptive multiple-shooting method proposed in this paper. We compare its performance with that of its peers, namely a method based on wavelet-collocation introduced in [11], a finite-difference scheme first analyzed in [3] and adopted here to solve multi-point SBVPs (see the Appendix for details of its derivation) and a simple-shooting method when it applies.

We have selected three test problems from the literature each exemplifying different characteristics of the solution process. The first problem is a 1-dimensional SBVP with a functional boundary condition and additive noise but the other two are linear 2-dimensional TP-SBVPs, the first with additive and the second with multiplicative noise. All of the algorithms are implemented in the MATLAB problem-solving environment and

executed on a core i5 processor, 2.4GHz, 4GB RAM computer.

Test Problem 1: In this numerical experiment, we try to solve the following 1-dimensional SBVP with functional boundary condition

$$\begin{cases} dX(t) = 1 \circ dW(t), & 0 \leq t \leq 1, \\ \int_0^1 X(s)ds = 0, \end{cases} \tag{4.1}$$

and having the exact solution [11]

$$X(t) = -\int_0^1 W(s)ds + W_t. \tag{4.2}$$

The integral boundary condition in (4.1) should be discretized (e.g. by the trapezoidal method) into a multi-point boundary condition of the form

$$\frac{\Delta\tau}{2}X(\tau_1) + \sum_{j=2}^{N_s-1} \Delta\tau X(\tau_j) + \frac{\Delta\tau}{2}X(\tau_{N_s}) = 0,$$

in which $\tau_j = (j-1)\Delta\tau$ for $j = 1, \cdots, N_s$ is the j-th switching point and $\Delta\tau = \frac{1}{N_s-1}$. We now place N_m equally-spaced points on the interval $I_i = [\tau_i, \tau_{i+1}]$ which act as the base mesh to integrate the resulting local SDE-IVPs and the global SDE problem. For each realization of the Wiener process (constructed on the base mesh), we solve the system of equations (3.1) for $\overline{\theta}_{\tau_j}$, $j = 1, 2, \cdots, N_s$ and interpolate them by (3.2) to arrive at a globally-defined piecewise linear initial approximation to the solution on the whole unit interval.

To find the location of shooting points on I_i, we start to synchronously integrate (3.3) and (3.4) on the base mesh with the schemes described in Section 3.1 to arrive at the first point satisfying (3.5) with $\alpha = 0$ and $\beta = 2.5$, from where we turn back to $\theta(t)$ and continue the process to reach τ_{i+1}. Similar procedure must be repeated for other intervals to find the set of all optimal shooting points on $[0, 1]$.

We are now ready to form and solve the nonlinear system (2.3) by a damped-Newton iteration (adopted from [9]) to obtain the optimal starting values at the shooting points $(\mathbf{s}_i, \, i = 1, 2, \cdots, N_s$ in (2.4)) and finally solve the original SDE with these initial values by the underlying (full) R3 scheme.

The accuracy of different schemes is measured via the measure E_∞ which is an average of the form

$$E_\infty := \frac{1}{M} \sum_{k=1}^M E(\omega_k)$$

over a fixed number M of realizations from the maximum grid-wise error

$$E(\omega_k) := \max_{i=1,2,\cdots,N} |\overline{X}(t_i, \omega_k) - X(t_i, \omega_k)|$$

approximating the expected supremum norm

$$\mathbb{E}\left(\|\overline{X}(t) - X(t)\|_\infty\right)$$

9

in which $\overline{X}(t)$ is our approximation and $X(t)$ is the exact solution. We have used the `quadl` function in MATLAB to approximate the integral term in the exact solution (4.2) which uses the adaptive Lobatto quadrature method.

The results of our computations are depicted in Tables 1 and 2. To compare the accuracy over a single realization ($M = 1$), we have provided Table 1 with columns reporting the global error (E_∞) for two different methods and a range of grid spacings in the problem domain. The meaning of N in the wavelet-based method is the number of collocation points and in the adaptive multiple-shooting method (or adaptive MSM for short) is the size of base grid used in the integration process. It must be noted that the number of switching points (N_s) we have used in the i-th row is chosen to be 2^i and the number of mid-points (N_m) is set accordingly. The superior accuracy of the proposed method (granting one order of magnitude more precision in the results) is obvious from this table. We have observed similar patterns of error behavior over many realizations ($M >> 1$) for the adaptive multiple-shooting but due to the unavailability of the data for the other scheme, we have not included them in the Table 4.

Table 1: Comparing the accuracy of the wavelet-based and adaptive multiple-shooting methods.

N	Wavelet-Collocation	Adaptive MSM
2^2	0.2058	0.0266
2^4	0.0997	0.0036
2^6	0.0075	0.0007

We also have compared the proposed method with a finite-difference scheme in Table 2. Here the errors are reported over $M = 500$ realizations (in both methods) and the finite difference equations are set up on all of the N grid-points of the base grid in our adaptive scheme. The column with the heading N_a indicates the average number of shooting points selected by the algorithm. Again we observe a higher accuracy for the adaptive method and a rapid rate of convergence to the exact solution.

Table 2: Comparing the accuracy of the finite-difference and adaptive multiple-shooting methods.

N	N_s	N_m	N_a	FD Method	Adaptive MSM
2^5	7	4	11	0.4819	0.0377
2^6	10	6	16	0.2728	0.0264
2^7	15	8	23	0.2477	0.0164
2^8	22	12	36	0.2409	0.0111
2^9	32	16	52	0.2287	0.0074

In order to investigate the rate of decay of the error (in the strong sense) for the adaptive multiple-shooting method, we have plotted Figure 4 which shows, in a logarithmic scale, the behavior of the global error in terms of increasing the number of switching points. One can observe that the rate of convergence is linear in $\Delta\tau$ and the line of linear regression applied to the data has a slope of $q = 1.0126$ with a residual $r = 0.0908$. This is a priori
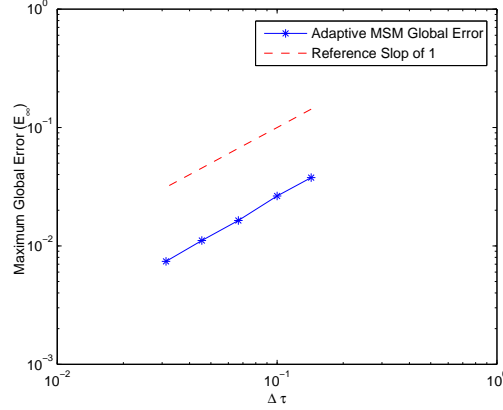
10

Figure 1: Asterisks: strong error measure for the adaptive multiple-shooting method applied to test problem (1). Dashed line: reference slope of 1.

anticipated as we have used a method of strong order of convergence one in the integration procedure and a super-linear convergent method in solving the set of nonlinear equations.

Test Problem 2: Here we solve the following 2-dimensional TP-SBVP

$$
\begin{cases}
dX(t) = (AX(t) + a)dt + (BX(t) + b) \circ dW_t, & 0 \le t \le 1, \\
H_0 X(0) + H_1 X(1) = c.
\end{cases}
$$

in which

$$
A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad a = \begin{bmatrix} 0 \\ c_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ c_2 \end{bmatrix}
$$

and

$$
H_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad H_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.
$$

Writing $X(t) = [X_1(t), X_2(t)]^T$, it could be shown that $X_1(t)$ solves a second-order SDE having the exact solution

$$
X_1(t) = c_1 \frac{t(t-1)}{2} + c_2(t-1) \int_0^t s \, dW(s) + t \int_t^1 (s-1) dW(s), \tag{4.3}
$$

and $X_2(t) = \frac{d}{dt} X_1(t)$ (see [3] for more details).

To obtain the initial trajectory $\theta(t) = [\theta_1(t), \theta_2(t)]^T$ for this test problem and supposing that a realization of $W(1)$ is simulated, we first solve the linear system

$$
\left(H_0 + H_1 + H_1 A + W(1) H_1 B \right) \overline{\theta}_{\tau_1} = c - H_1 a - W(1) b
$$

for $\overline{\theta}_{\tau_1}$ and then solve another linear system

$$
H_1 \overline{\theta}_{\tau_2} = c - H_0 \overline{\theta}_{\tau_1}
$$

for $\overline{\theta}_{\tau_2}$. Now we use linear interpolation to obtain $\theta(t)$ over the whole unit interval. In checking the relation (3.5), we have used the l_∞-norm on both sides with $\alpha = 2$ and

11

$\beta = 1.5$. We also compute the integral terms in the exact solution (4.3) by a highly-accurate trapezoidal scheme.

We use the finite-difference and also the simple-shooting methods as two competing approaches to solve this same problem. Table 3 summarizes our computational results for the case $c_1 = 1$ and $c_2 = 1$ averaged over $M = 1000$ realizations. We can observe that

Table 3: Comparing the accuracy of the finite-difference, simple-shooting and adaptive multiple-shooting methods.

N	N_a	FD Method	SS Method	Adaptive MSM
2^5	9	0.0121	0.0123	1.14e-16
2^6	11	0.0064	0.0065	1.70e-16
2^7	13	0.0032	0.0032	2.59e-16
2^8	15	0.0016	0.0016	4.62e-16
2^9	17	0.0008	0.0008	9.62e-16

while both finite-difference and simple-shooting methods converge uniformly to each other (in terms of accuracy and order of convergence), the adaptive multiple-shooting beats them and gives very accurate results. We also observe a steady growth in the errors as we increase N which could be attributed to the accumulation of round-off errors in the solution process.

In order to show the efficiency of the adaptive method in the weak sense and using the fact that we can compute the expectation of the exact solution and its non-central second moment by the following formulas

$$\mathbb{E}(X_1(t)) = c_1 \frac{t(t-1)}{2}, \quad \mathbb{E}(X_1^2(t)) = (3c_1^2 + 4c_2^2)\frac{t^2(t-1)^2}{12}, \tag{4.4}$$

we have approximated these expected values on a range of points in the solution domain by averaging over $M = 10000$ realizations of the solution process (computed pointwise) and have compared the results with that of other schemes listed in Table 4. While all methods have a comparable accuracy, the performance of the adaptive method is actually slightly better at all points in the range $[0, 1]$.

Table 4: Comparing the accuracy in the weak-sense of the Heun simple-shooting, finite-difference and adaptive multiple-shooting methods.

t	Heun Simple-Shooting [3] $E(X(t))$	$E(X^2(t))$	FD Method [3] $E(X(t))$	$E(X^2(t))$	R3 Adaptive MSM $E(X(t))$	$E(X^2(t))$	Exact $E(X(t))$	$E(X^2(t))$
0.0	-0.0000	0.00000	-0.0000	0.00000	-0.0000	0.0000	-0.0000	0.0000
0.2	-0.0800	0.01499	-0.0805	0.01497	-0.0800	0.0151	-0.0800	0.0149
0.4	-0.1194	0.03338	-0.1201	0.03357	-0.1199	0.0338	-0.1200	0.0336
0.6	-0.1192	0.03346	-0.1193	0.03328	-0.1202	0.0338	-0.1200	0.0336
0.8	-0.0793	0.01486	-0.0791	0.01472	-0.0800	0.0150	-0.0800	0.0149
0.1	-0.0000	0.00000	-0.0000	0.00000	-0.0000	0.0000	-0.0000	0.0000

Test Problem 3: As the last example, we solve the 2-dimensional SDE-BVP system

(adopted from [24]) of the form:

$$\begin{cases} dX(t) = B_1 X(t) \circ dW_1(t) + B_2 X(t) \circ dW_2(t), & 0 \le t \le 1, \\ H_0 X(0) + H_1 X(1) = c, \end{cases}$$

in which

$$B_1 = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$

and

$$H_0 = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \quad H_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

This equation has an exact solution of the form

$$X(t) = \begin{bmatrix} e^{W_1(t)} \left(1 - e^{-W_2(1)} + \alpha_t^0 e^{-W_2(1)} \right) \\ e^{W_2(t)} - e^{W_2(1)} \end{bmatrix} \tag{4.5}$$

where

$$\alpha_t^s = e^{W_1(t)} \int_s^t e^{-W_1(u)} e^{W_2(u) - W_2(s)} dW_1(u). \tag{4.6}$$

Similar to test problem (2), we could obtain the initial trajectory $\theta(t) = [\theta_1(t), \theta_2(t)]^T$ by first simulating a realization from $W(1) = [W_1(1), W_2(1)]^T$ and then solving the two linear systems

$$\left(H_0 + H_1(I + B_1 W_1(1) + B_2 W_2(1)) \right) \overline{\theta}_{\tau_1} = c, \tag{4.7}$$

$$H_1 \overline{\theta}_{\tau_2} = c - H_0 \overline{\theta}_{\tau_1} \tag{4.8}$$

for $\overline{\theta}_{\tau_1}$ and $\overline{\theta}_{\tau_2}$ respectively. Now $\theta(t)$ is computed by linear interpolation and the integration is started to obtain the location of shooting points in the base grid. We use the absolute values of the second components of $\widehat{X}$, $\widetilde{X}$ and θ in (3.5) with $\alpha = 1.5$ and $\beta = 2$ and approximate the integral terms in (4.5) and (4.6) by a sufficiently accurate trapezoidal scheme.

The results of our computations for this test problem are reported in Table 5. For comparison purposes, we have also included the results of applying fixed-step multiple-shooting method in this table. To be fair in the competition, we have selected the number of shooting points in the fixed-step multiple-shooting equal to the average number of adaptive shooting pointes (N_a) selected by the adaptive algorithm.

In order to examine the strong order of convergence of the adaptive scheme, we have prepared Figure 4 which shows clearly (and in a logarithmic scale) that this order is one. The result of linear regression applied to the data used in the figure gives us a slope of $q = 1.0515$ with residual $r = 0.1190$ which is acceptable.

5   Concluding Remarks

The numerical solution of boundary value problems in stochastic differential equations is a highly unexplored territory of the SDE world requiring the special attention of the experts in the field to devise methods of high accuracy and efficiency with low computational

Table 5: Comparing the accuracy of the fixed-step multiple-shooting and adaptive multiple-shooting methods.

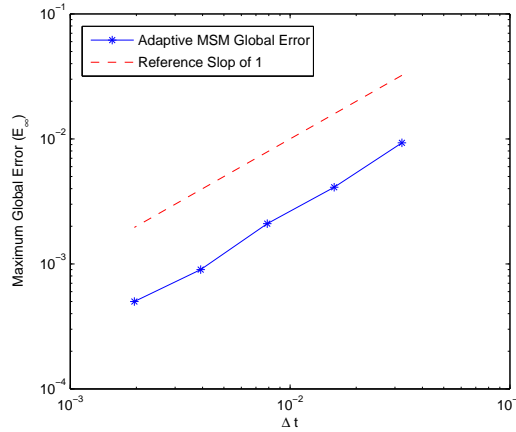| N	N_a	Fixed MSM Method	Adaptive MSM R3
2^5	3	0.0137	0.0093
2^6	4	0.0049	0.0041
2^7	4	0.0025	0.0021
2^8	5	0.0012	0.0009
2^9	5	0.0006	0.0005

Figure 2: Asterisks: strong error measure for the adaptive multiple-shooting method applied to test problem (3). Dashed line: reference slope of 1.

demand and complexity. We have proposed in this paper, an adaptive multiple-shooting method for general multi-point SBVPs based on a stochastic Runge-Kutta integrator. Although the adaptation criteria is simple and easily implementable in the method, it gives acceptable results in comparison with some other non-adaptive alternatives proposed in the literature. The next step in our research (as explained briefly in Section 3) is to make use of more elaborate stopping criteria in the selection of shooting points and its theortical analysis. We could also incorporate the idea of adptive time-stepping in the integration process itself which we anticipate to improve the accuracy further but needs a theoretical foundation to prove the stability of the overall scheme in a unified manner. Finally, we must also mention the need for introduction of nonlinear test instances into the field which is of great importance for testing and benchmarking purposes in the algorithmic developements expected to be seen in the near future.

A   Finite-Difference Method for Multi-Point SBVPs

In this appendix, we present a finite-difference scheme for multi-point SBVPs of the form

$$L[X](t) = dW(t) \tag{A.1}$$

in which the operator L is defined by

$$L = D^n + a_{n-1}(t)D^n + \cdots + a_1(t)D^1 + a_0(t), \quad D := \frac{d}{dt},$$

where the coefficients $a_i(t)$'s, $i = 0, 1, \cdots, n-1$ are continuous functions defined on $[0,1]$. We also append (A.1) with boundary conditions of the form

$$\sum_{j=1}^{N_s} \alpha_{ij} X(\tau_j) = c_i, \qquad 1 \le i \le n, \tag{A.2}$$

defined on some switching points $0 \le \tau_j \le 1$, $j = 1, 2, \cdots, N_s$ (see [2] for a detailed study of some features of the solution to these problems).

Similar to ordinary differential equations, the SBVP (A.1)-(A.2) can be turned into a first order system

$$d\mathbf{Y}(\mathbf{t}) + A(t)\mathbf{Y}(\mathbf{t}) = d\mathbf{W}(\mathbf{t}), \tag{A.3}$$

constrained to satisfy

$$\sum_{j=1}^{N_s} \alpha_{ij} Y_n(\tau_j) = c_i, \qquad 1 \le i \le n, \tag{A.4}$$

in which $\mathbf{Y}(\mathbf{t}) = (Y_1(t), \cdots, Y_n(t))$, $Y_i(t) = D^{n-i}X(t)$ for $1 \le i \le n$, $\mathbf{W}(t) = (W(t), 0, \cdots, 0)$ and

$$A(t) = \begin{bmatrix} a_{n-1}(t) & a_{n-2}(t) & \cdots & a_1(t) & a_0(t) \\ -1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 0 \end{bmatrix}.$$

To solve (A.3)-(A.4) by the finite-difference method, we first construct a base mesh including the switching points of the form

$$0 = t_1 < t_2 < \cdots < t_{N-1} < t_N = 1$$

and use an explicit one-step difference scheme on this mesh to arrive at

$$\mathbf{Y}^{j+1} - \mathbf{Y}^j + A^j \mathbf{Y}^j = \Delta \mathbf{W}^j,$$

in which $\mathbf{Y}^j = [Y_1^j, Y_2^j, \ldots, Y_n^j]^T$ is an approximation to $\mathbf{Y}(t_j)$, $A^j := A(t_j)$ and $\Delta \mathbf{W}^j := \mathbf{W}(t_{j+1}) - \mathbf{W}(t_j)$. Simplifying the above relation we obtain

$$\mathbf{Y}^{j+1} + (A^j - I)\mathbf{Y}^j = \Delta \mathbf{W}^j$$

and arranging them in a sequential manner (into a linear structure) we reach to

$$\Lambda \tilde{\mathbf{Y}} = \mathbf{w} \tag{A.5}$$

in which

$$\Lambda_j = A_j - I, \quad \Lambda = \begin{bmatrix} \Lambda_1 & I & & & \\ & \Lambda_2 & I & & \\ & & \ddots & \ddots & \\ & & & \Lambda_{N-1} & I \\ \Phi_1 & \Phi_2 & \cdots & \Phi_{N-1} & \Phi_N \end{bmatrix}_{(N \times n) \times (N \times n)}$$

15

and

$$\Phi_j := \begin{bmatrix} 0 & 0 & \dots & 0 & \alpha_{1j} \\ 0 & 0 & \dots & 0 & \alpha_{2j} \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & \alpha_{nj} \end{bmatrix} = \begin{bmatrix} \alpha_{1j} \\ \alpha_{2j} \\ \vdots \\ \alpha_{nj} \end{bmatrix} \times [0, 0, \dots, 0, 1]_{1 \times n}.$$

Also the vector $\tilde{\mathbf{Y}}$ has the form

$$\tilde{\mathbf{Y}} = [\mathbf{Y}^1, \mathbf{Y}^2, \dots, \mathbf{Y}^N]^T$$

and

$$\mathbf{w} = [\Delta \mathbf{W}^1, \Delta \mathbf{W}^2, \dots, \Delta \mathbf{W}^N]^T.$$

By solving (A.5) for each realization of the Wiener process, one obtains the corresponding realization for the solution process on the base mesh which is what we have reported for test problem (1).

References

[1] Ahn, H. & Kohatsu-Higa, A. (1995), *The Euler scheme for anticipating stochastic differential equations*, Stochastics and Stochastics Reports, **54**, 247–269.

[2] Alabert, A. & Ferrante, M. (2003), *Linear stochastic differential equations with functional boundary conditions*, The Annals of Probability, **31**, 2082–2108.

[3] Allen, E. J. & Nunn, C.J. (1995), *Difference methods for numerical solution of stochastic two-point boundary-value problems*, in Proceedings of the First International Conference on Difference Equations, (San Antonio), Elaydi, S. N., Graef, J.R., Ladas, G. & Peterson, A. C. (editors), 17–28, Gordon and Breach Publishers, Amsterdam.

[4] Arciniega, A. (2007), *Shooting Methods for Numerical Solution of Nonlinear Stochastic Boundary-Value Problems*, Stochastic Analysis and Applications, **25**, 1, 187 - 200.

[5] Arciniega, A. & Allen, E. (2004), *Shooting methods for numerical solution of stochastic boundary-value problems*, Stochastic Analysis and Applications, **22**, 5, 1295–1314.

[6] Ascher, U. M., Mattheij, R. M. M. & Russel, R. D. (1995), *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, SIAM, Philadelphia.

[7] Burrage, K. & Burrage, P. M. (1998), *General order conditions for stochastic Runge-Kutta methods for both commuting and non-commuting stochastic ordinary differential equation systems*, Applied Numerical Mathematics, **28**, 16, 161–177.

[8] de Hoog, F. R. & Mattheij, R. M. M. (1987), *An algorithm for solving multi-point boundary value problems*, Computing, **38**, 219–234.

[9] Deuflhard, P. (1975), *A relaxation strategy for the modified Newton method*, in Optimization and Optimal Control, Bulirsch, R., Oettli, W. & Stoer, J. (editors), 59–73, Lecture Notes in Mathematics, Vol. 477, Springer, New York.

[10] England, R. & Mattheij, M. M. R. (1984), *Sequential step control for integration of two-point boundary value problems*, Numerical Analysis: Proceedings of the Fourth IIMAS Workshop, (Mexico), Hennart, J. P. (editor), 221–234, Springer-Verlag, Berlin.

[11] Esteban-Bravo, M. & Vidal-Sanz, J. M. (2006), *Valuation of boundary-linked assets by stochastic boundary value problems solved with a wavelet-collocation algorithm*, Computers and Mathematics with Applications, **52**, 137–160.

[12] Ferrante, M., Kohatsu-Higa A. & Sanz-Solé M. (1996), *Strong approximations for stochastic differential equations with boundary conditions*, Stochastic Processes and their Applications, **61**, 323–337.

[13] Foroush Bastani, A. & Hosseini, S. M. (2007), *A new adaptive Runge-Kutta method for stochastic differential equations*, Journal of Computational and Applied Mathematics, **206**, 631–644.

[14] Fouque, J. P. & Merzbach, E. (1994), *A limit theorem for linear boundary value problems in random media*, Annals of Applied Probability, **4**, 2, 549–569.

[15] Garnier, J. (1995), *Stochastic invariant imbedding. Application to stochastic differential equations with boundary conditions*, Probability Theory and Related Fields, **102**, 249–271.

[16] Keller, H. B. (1976), *Numerical Solution of Two Point Boundary Value Problems*, SIAM, Philadelphia.

[17] Kloeden, P. E. & Platen, E. (1992), *The Numerical Solution of Stochastic Differential Equations*, Springer-Verlag, Berlin.

[18] Krener, A. (1986), *Reciprocical processes and the stochastic realization problem for a causal system*, In: Byrnes, C.I. & Lindquist, E. (eds.), Modelling, Identification, and Robust Control, Elsevier, Amsterdam.

[19] McDonald, S. (2006), *Finite difference approximation for linear stochastic partial differential equations with method of lines*, Unpublished Report, Munich Personal RePEc Archive. (http://mpra.ub.uni-muenchen.de/3983/).

[20] Milstein, G. N. (1995), *Numerical Integration Of Stochastic Differential Equations*, Kluwer, Dordrecht.

[21] Nualart, D. (1995), *The Malliavin Calculs and Related Topics*, Springer-Verlag, Berlin.

[22] Nualart, D. & Pardoux, E. (1991), *Boundary value problems for stochastic differential equations*, The Annals of Probability, **19**, 1118–1144.

[23] Nualart, D. & Pardoux, E. (1991), *Second order stochastic differential equations with Dirichlet boundary conditions*, Stochastic Processes and Its Applications, **39**, 1–24.

[24] Ocone, D. & Pardoux, E. (1989), *Linear stochastic differential equations with boundary conditions*, Probability Theory and Related Fields, **82**, 489–526.

[25] Ocone, D. & Pardoux, E. (1998), *Linear stochastic integrals and the Malliavin calculus*, Probability Theory and Related Fields, **82**, 439–526.

[26] Pardoux, E. (1986), *Equations du lissage non-lineaire*, In: Korezlioglu, H., Mazziotto, G. & Szpirglas, J. (eds.), Filtering and Control of Random Processes. (Lect. Notes Control Inf. Sci., vol. 61), Springer, Berlin.

[27] Prigarin, S. M. & Winkler, G. (2002), *Numerical solution of boundary value problems for stochastic differential equations on the basis of the Gibbs sampler*, Preprint **02-27**, GSF Neuherberg, 6 pages.

[28] Roberts, S. M. & Shipman, J. S. (1972), *Two-Point Boundary Value Problems: Shooting Methods*, American Elsevier, New York.

[29] Stoer, J. & Bulirsch, R. (1993), *Introduction to Numerical Analysis*, Springer-Verlag, New York.

[30] Zeitouni, O. & Dembo, A. (1987), *A maximum a-posteriori estimator for trajectories of diffusions*, Stochastics, **20**, 341, 221–246.

[31] Zeitouni, O. & Dembo, A. (1990), *A change of variables formula for Stratonovich integrals and existence of solutions for two-point stochastic boundary value problems*, Probability Theory and Related Fields, **84**, 411–425.